

HARALD NAHRSTEDT

# C++ für Ingenieure

---

## Einführung in die objektorientierte Programmierung

### **C++ Quellcodes zum Buch Kapitel 5**

Erstell am 27.04.2009

Beschreibung Diese Seiten enthalten den Quellcode aus dem Buch. Sie können kopiert und zur eigenen Programmierung weiter verwendet werden. Für die Ausführung wird jedoch keine Haftung übernommen.

**Code 5-1** Die generische Implementierung von *max*

```
template <class T>
T max(T a, T b)
{
    if (a < b)
        return b;
    else
        return a;
}
```

**Code 5-2** Aufruf des Funktionstemplates *max*

```
int main()
{
    cout << max (33, 44) << endl;
    system("Pause");
}
```

**Code 5-3** Aufruf des Funktions-Templates *max* mit expliziter Typangabe für den Rückgabewert

```
int main()
{
    cout << max<int> (33, 44) << endl;
    system("Pause");
}
```

**Code 5-4** Funktions-Template *swap*

```
/* GenSwap.cpp
   Dieses Beispiel zeigt eine
   generische Vertauschungs-Funktion
*/

#include <iostream.h>

template <typename X>
void swap (X& a, X& b)
{
    X temp = a; // temp mit a initialisiert
    a = b;
    b = temp;
}

int main()
{
    int a=33, b=44;
    cout << "vorher: " << a << ", " << b << endl;
    swap (a, b);
    cout << "nachher: " << a << ", " << b << endl;
    system("Pause");
}
```

```
}
```

**Code 5-5** Sortierung von Vektoren

```
/* GenSort.cpp
   Dieses Beispiel sortiert Vektoren
   beliebiger Länge und Typs
   mit Hilfe von Templates
*/

#include <iostream.h>

template <typename X>
void swap (X& a, X& b)
{
    X temp = a;  // temp mit a initialisiert
    a = b;
    b = temp;
}

template <class T, int vector>
void sort (T (&a)[vector])
{
    for (int i=0; i<vector; ++i)
        for (int j=i+1; j<vector; ++j)
            if (a[j] > a[i])
                swap (a[i], a[j]);
}

int main()
{
    int i;
    int    v1[12] = {3,9,4,2,8,8,1,7,5,11,10,6};
    float  v2[6]  = {0.4, 3.1, 5.4, 6.7, 0.4, 0.42};
    sort (v1);
    sort (v2);
    cout << "v1: " << endl;
    for (i=0;i<12;i++) cout << v1[i] << endl;
    cout << "v2: " << endl;
    for (i=0;i<6;i++) cout << v2[i] << endl;
    system("Pause");
}
```

**Code 5-6** Hauptprogramm mit expliziter Angabe der Template-Parameter

```
int main()
{
    int i;
    int    v1[12] = {3,9,4,2,8,8,1,7,5,11,10,6};
    float  v2[6]  = {0.4, 3.1, 5.4, 6.7, 0.4, 0.42};
    sort<int> (v1);
    sort<float> (v2);
    cout << "v1: " << endl;
    for (i=0;i<12;i++) cout << v1[i] << endl;
```

```
    cout << "v2: " << endl;
    for (i=0;i<6;i++) cout << v2[i] << endl;
    system("Pause");
}
```

**Code 5-7** Hauptprogramm mit explizite Angabe zur Nutzung des max-Templates

```
int main()
{
    int i = 33;
    float f = 44.44;
    cout << max<float> (i, f) << endl;
    system("Pause");
}
```

**Code 5-8** Template zur Erzeugung von Stack-Mustern

```
/* GenStack.h
   Dieses Beispiel zeigt
   die Definition einer
   generischen Klasse Stack
*/

#ifndef STACK_H
#define STACK_H

template<int size, class Elem>
class Stack
{
// Parameter
private:
    Elem space [size];
    int index;
// Methoden
public:
    Stack();
    void push (Elem e);
    void pop  (Elem &e);
    void out  ();
};

#endif
```

**Code 5-9** Methoden für Stack-Muster

```
/* GenStack.cc
   Dieses Beispiel zeigt
   die Methoden einer
   generischen Klasse Stack
*/

#include "GenStack.h"
```

```
// Konstruktor
// Stack<size, Elem> bezeichnet die Klasse
// zu der der Konstruktor Stack gehört
template<int size, class Elem>
Stack<size, Elem>::Stack (): index(-1)
{}

// Einfügen
template<int size, class Elem>
void Stack<size, Elem>::push (Elem e)
{
    if (index < size)
    {
        index += 1;
        space[index] = e;
    }
}

// Entfernen
template<int size, class Elem>
void Stack<size, Elem>::pop (Elem &e){
    if (index >= 0)
    {
        e = space[index];
        index -= 1;
    }
}

// Ausgabe
template<int size, class Elem>
void Stack<size, Elem>::out ()
{
    int i;
    cout << "Index = " << index << endl;
    cout << "Stack = ";
    for (i=0;i<=index;i++)
    {
        cout << space[i] << "\t";
    }
    cout << endl;
}
```

**Code 5-10** Hauptprogramm für die Nutzung eines int-Stacks mit 9 Elementen

```
/* GenStack.cpp
   Dieses Beispiel zeigt
   die Nutzung einer
   generischen Klasse Stack
*/

#include <iostream.h>
#include "GenStack.cc"

int main()
```

```
{  
    int a;  
  
    Stack<9,int> intStack;    // Instanziierung des Templates  
  
    intStack.push(11);  
    intStack.out();  
    cout << "--- 1 ---" << endl;  
  
    intStack.push(22);  
    intStack.out();  
    cout << "-- 2 ---" << endl;  
  
    intStack.pop(a);  
    intStack.out();  
    cout << "Entfernt: " << a << endl;  
    cout << "-- 3 ---" << endl;  
  
    system("Pause");  
}
```